THE DESIGN AND DEVELOPMENT OF AN ONLINE DATABASE FOR THE CAREER SERVICES DEPARTMENT OF THE UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

by
Hope McCullough

A Master's paper submitted to the faculty of the School of Information and Library Science of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in Information Science.

Chapel Hill, North Carolina

July, 2000

Approved by:

_____
Advisor

CONTENTS

Section

BACKGROUND

Purpose of the Project and Current Environment of the System

The purpose of this project is to streamline the data management process associated with career events sponsored by the University Career Services (UCS) Office of the University of North Carolina at Chapel Hill.

Every year the Career Services Office sponsors the following three Career Events in late September:

- Technology Career Fair
- Minority Career Night
- Carolina Career Fair

The goal of these events is to allow UNC students and prospective employers to meet one another. This provides students with the opportunity to learn more about the companies they are interested in, and employers are able to recruit students for available positions. Employers pay a fee to the UNC Career Services Office in order to participate in these events. Non-profit companies are offered a reduced fee.

The current information for the three different job fairs is stored in an Excel spreadsheet (appendix A). There is a separate datasheet for each Career Event; a sheet containing contact names and addresses for each company; and a sheet containing contact names, phone numbers, email addresses, and website addresses for each company.

The sheet with the addresses is used to create mailing labels for the registration forms that are sent out in early June (appendix B). Once the registration forms are filled out and

returned with payment by the employers, the appropriate information is entered in the Excel sheet corresponding with each career event. If contact information is changed, the new information is entered into the appropriate contact information spreadsheet.

Problems with the Current System

This system was adequate when there was only one career event to keep up with, the Carolina Career Fair, but with the addition of the Technology and Minority Career events, data entry and upkeep quickly became a headache. In order to view information about which fairs a certain company will be attending, it was necessary to shuffle between the three different datasheets for each fair. In addition to this, the contact names are duplicated in the two datasheets that contain contact information. This can lead to problems with data integrity.

Another problem with the current system is the amount of time required to enter all of the data from the registration forms. The employers must fill out the forms by hand and return them to the Career Services office. Someone from the Career Services office then must enter the information from the forms into the Excel spreadsheet.

In addition to this, before students can view the list of employers who will be attending a specific job fair, the company's data must be transcribed from the returned registration forms into the Excel spreadsheet. Once this data is entered, web pages must be manually created from the data in the Excel file. This is a tedious process and accounts for a considerable delay between the time employers are registered for a career event and the time the list of employers coming to each event is updated online.

Goals of the Proposed System

The goal of this project was to create an internet-accessible relational database. This will allow employers to enter their information online and UNC students to search for up to date information on which companies are coming to which fairs. The database provides more flexibility in using and viewing the data from an administrative perspective in the Career Services office. The ability to have employers enter their registration information via the web will greatly decrease the data-entry time required by employees at Career Services. In addition to this, the lag time between data entry and student access to pertinent information will be greatly reduced.

Users of the Proposed System

There are three sets of users for the proposed online career database. The first set of users is the employees at Career Services. They need to be able to access the database to perform administrative tasks. These tasks include the following:

- Entering new companies in the database along with contact information (company name, contact first name, and contact last name). This must be done before employers can access and update their record online since the password is derived from the contact's first and last name. See section titled "Employer Web Interface Design" for a more detailed discussion of the password.
- Assigning an event code and invoice number to each company. This information is for billing purposes and must be assigned before companies can

register online since the employers are required to write this information on their payment.

- Entering values for "date_form_received", "date_fee_received", "amount_received_for_fairs", and "amount_received_for_lists" fields. The "date_fee_received" and "amount_received_for_fairs" fields are used to determine which companies are officially registered for fairs.
- Updating "registration_confirmed" field. Once the full amount owed for the fairs is received, the companies are counted as officially registered for the fairs they signed up for and a space is reserved for them.
- Editing fair registration information if a representative of an organization calls to make changes after online form has been submitted.
- Running queries to find out:
  - ◊ How many lunches were requested for Technology and Carolina Fairs?
  - ◊ How many representatives are coming for each event?
  - ◊ How many parking spaces are needed?
  - ◊ How many student lists need to be run off?
  - ◊ What other lists were requested?
  - ◊ How many companies are non-profit?
  - ◊ What accommodations were requested by which companies?
  - ◊ Which companies are confirmed for which fairs?
  - ◊ Which companies have registered online but have not returned registration fee in mail?
  - ◊ Which companies owe money?

- Running reports to:
  - ◊ Create confirmation letters. The report will state which fairs the company is registered for.
  - ◊ Create student handouts to be passed out at fairs. The handouts list companies ordered by industry. The following is listed for each organization: fairs company is participating in, organization name, contact name, address, website address of company, majors considered, job titles, and organization profile (appendix C.)

There are two people in the UNC Career Services office who will be given the necessary permissions to view and edit the database. These two users will likely access the database on a daily basis during the several months prior to the career events. For the rest of the year, they will access the database periodically to update contact information and to run queries during the planning process for the next year's career events. For example, in the process of planning the next year's events they might want to know how many representatives were registered for each event in the previous year.

The second set of users is the representatives of organizations that will be registering for and attending the three different career events. These users need online access to the registration form so they can enter their contact information and register for the various fairs online.

They will be sent a brochure in early June that lists the url for the online registration form along with the password they will be required to enter to access the form. Once they submit their registration form, they will no longer be able to access the online database but must instead contact the UNC Career Services office directly to make

any changes (e.g. to add representatives). The reason for disallowing subsequent registration entries online is to avoid confusion. Since the system is new, it is important to keep the complexity to a minimum. Once the current system is accepted, it will be possible to allow employers to change their entries up to a certain date or until the time that their payment is received and processed. If only a small number of employers call to make changes, however, it may not be necessary to expend the resources to add this functionality.

There are approximately 650 companies whose contact information is stored in the database and who are sent an invitation to register for the career fairs. Thus the maximum amount of users possible in this group is approximately 650 and the frequency of use for each individual user is one use.

The third set of users is UNC students. They need to be able to search the database via an online interface so they can find out which companies are coming to which career events. They also need to be able to view information about those companies, such as: contact name, contact address, website address of company, majors considered, job titles of available positions, and a description of the organization. This information also will be provided to students in handouts that will be passed out at the career fairs (appendix C).

Why a Web Database?

There are many reasons for implementing a web database. A relational database, regardless of whether it was accessible online or not, was required because of the growing complexity of the information that was being stored. The current system of

storing information in an Excel spreadsheet made manipulating the data a confusing task and presented the possibility of inconsistent data since contact information was duplicated across datasheets.

One possibility would have been to design an Access database to replace the current Excel spreadsheet. Registration forms would still be sent through the mail and filled out by employers. Once the registration forms were returned to the Career Services office, someone would need to manually transfer the data from the registration forms into the Access database. This redundancy in data entry not only requires extra time for Career Services employees to enter the data, but also increases the possibility for errors in the database.

By connecting the database to the Web and allowing employers to enter their registration information online, this problem of redundant data entry is eliminated. This does not completely eliminate the possibility of inconsistent data, however, since employers could still make errors in data entry. In order to minimize this possibility, data validation checks were built into the online forms.

In addition to reducing redundancy in data entry, the online database makes data entry easier for employers. Information that remains the same from one year to the next would not have to be filled out every year by the employers. For example, if the contact person for a company is the same as the previous year and their address has not changed, the online form would simply pull up this information from the database. The person filling out the online form also would not need to compose a new organization description every year but could simply edit the one currently stored in the database.

This may decrease employer response time since it significantly lowers the effort required to fill out the form.

Another advantage of having a web accessible database is that it makes it possible for students to have quicker access to the information about which companies are coming to which job fairs. An online database will make it possible to create pages that will be automatically updated from the database. This will provide students with more up to date information and greatly decrease the amount of time required to update the web pages for students to access.

A web database may also present some disadvantages that must be weighed against the advantages. One potential problem is employer acceptance of an online registration process. Employers may be reticent to submit their registration information online. This would be even more likely to be a problem if online payment were involved. Since UNC policy does not allow online payment, however, that will not come into play with this project. We have devised an alternative means of coping with the inability to accept online payments. For a detailed description of the proposed registration process, see the section titled "Employer Web Interface Design".

To deal with the possibility that employers may be hesitant to accept the new system, a phone number will be provided on the initial log-in screen that employers can call if they have any questions about or problems with the new system. An Adobe Acrobat PDF version and Microsoft Word version of the registration form will be available if employers call with problems or would simply prefer to fill out a paper form. They will need to get their invoice number from someone in Career Services then they

can either have the Word or PDF version of the form emailed to them or download and print a copy from the Web.

Security is another issue that must be considered when implementing an online database. To protect the online database against unauthorized changes, the url for the registration form will be given out only to companies who are invited to the career events. Each company also will be given a password they must provide in order to access and fill out the online form (See section titled "Employer Web Interface Design" for a more detailed description of the password.)

DESIGN DECISIONS

Choosing a Technology

Once the decision was made to develop a web accessible database, a technology needed to be chosen from among the many options available. Microsoft's Access was chosen as the relational database management system (RDBMS) that would be used to create the database and the administrative interface. The rationale for using Access as the RDBMS was the fact that the software was already installed on the computers at Career Services and the employees there were already somewhat familiar with the software.

The other choice would have been to use Microsoft's SQL Server. SQL Server is also available on one of the machines at Career Services, but someone would need to be trained to maintain the software. While SQL Server is much more powerful than Access, it is also more complex to administer.

Once the RDBMS was chosen, the next step was choosing a "glue" technology to connect the database to web accessible pages. Active Server Pages (ASP) technology was chosen for this purpose. The rationale for using ASP was the fact that the web server at Career Services, a Microsoft NT Server running Internet Information Server, supports ASP extensions. Therefore, no extra software needed to be installed, as would have been the case with Allaire's Cold Fusion. Also taken into consideration was the future pool of skilled workers that would be necessary to maintain the online database.

Currently, Microsoft is making a big push in the E-Commerce arena. As a result of this, many jobs are available for ASP developers and students in Information Science and Computer Science are eager to learn how to program and support ASP. This ensures that the Career Services department will be likely to find a student in the UNC-Chapel Hill community who could support the online database in the future.

Load Concerns

The choice of Access as the RDBMS brought up some questions about possible load concerns. The maximum number of concurrent users that Access can handle is 255. Performance greatly degrades, however, if you have over 20 simultaneous users. (Kauffman 1999, 655) For the online registration form, there is not a great chance that more than 20 users will access the database simultaneously. Because of this, the advantages of using Access outweighed the concerns about possible performance problems under heavy loads.

There is a possibility that there may be some performance problems with the student interface, especially in the time immediately before the fairs. Server traffic can be

monitored once the system is in place, however, and if performance visibly suffers with Access, there are two alternatives. The first alternative is to migrate the database to SQL Server. The advantage of this solution is that SQL Server was designed with the web in mind and is, therefore, much more powerful than Access when it comes to supporting multiple users. The disadvantage of this solution, as mentioned earlier, is the expense and time required to train someone in Career Services to maintain the database. For the time being, this expense outweighs any advantages, and migrating to SQL Server is viewed as a last resort.

The second alternative is to write a script that will create static pages from a dynamic ASP page. Instead of allowing students to query the database directly, the static pages can be updated on a regular basis by someone in the Career Services office. The advantage of this solution is that it eliminates calls to the database from the student interface, which will greatly improve performance. The disadvantage of this system is that it requires extra time for someone in career services who would have to routinely update the student pages. This second alternative is the one that will be put in place if there are any performance problems once the student interface is implemented. Having this alternative already planned will enable a fast response if a problem actually does present itself.

DEVELOPMENT PROCESS

Information Gathering

The first step in creating the new system was meeting with the UCS Assistant Director and a Secretary at UCS, to find out about the current system in place for gathering and manipulating the registration information. These two women are in charge of organizing and coordinating the three career events. I worked with them to find out what information was gathered, how it was gathered, and who was involved in the process.

The information gathered by the UCS office is that listed on the current registration form (appendix B). This information is currently gathered by sending hard copies of the registration form to employers in early June. The employers fill out and return their registration form with payment.

The Secretary at UCS in charge of the career fairs processes these forms, entering the necessary information in the Excel spreadsheet mentioned earlier. In addition to entering the information from the returned registration forms, she also enters the date the form was received, the date payment was received, the amount received for the fairs, the amount received for student lists, and the amount owed if full payment was not received. Once payments are processed, companies that have paid in full are sent a confirmation letter listing which fairs they are registered for along with dates and times of fairs and information about who to contact with any questions.

The main people involved in gathering and processing this information are the Assistant Director of UCS and the Secretary in charge of the fairs. I also found out how

the data was currently used and we discussed the types of reports and queries that might be needed in the future. See the section titled "Users of Proposed System" for a more detailed description of reports and queries that will be needed.

Database Design

The next step was to design the database. I created an Entity-Relationship (ER) diagram to use as a blueprint for the database (appendix D). The requirements gathered in the information section motivated the design of the database. Once the ER diagram was complete, I used it as a guide in designing the actual tables and relationships that would make up the Access database (appendix E).

Employer Web Interface Design

Once the database was fleshed out, I began discussing the employer web interface with the two people at UCS in charge of the career fairs. I used their current registration form as a template for the online interface (appendix B). The first step was creating a process-flow diagram (appendix F). The process-flow diagram modeled the actions employers would take to access the registration form and to enter and submit their data. One question that came up in the process-flow diagram was how to allow employers to access their data. One choice would be to allow employers to choose their company from a pull-down list. This would essentially allow anyone to update any employers information. This was unacceptable so a decision was made to use a password.

The next step was deciding on how to create the password. I originally created a Perl script that generated a list of seven-character random passwords. The advantage of this

solution was the fact that the passwords would be more difficult to hack into. The disadvantage of this solution was the fact that the passwords would be very difficult for employers to remember. The two people in charge of maintaining the database expressed concerns that employers would lose their password and have to call the UCS office to find out what it was. This would increase the burden on resources at UCS rather than alleviating it.

They were also concerned about the difficulty of assigning new passwords when they needed to add a new company to the database. Both of these concerns were valid and I tried to devise an algorithm for creating passwords that would be easy to create and easy to remember.

The solution I came up with was to use the contact person's first and last name to generate a password. Each contact would be assigned a password in the following format: "first name" + "*" + "last name". So, for instance, if the contact's first name was John and his last name was Doe, his password would be "John*Doe". The passwords are case-sensitive.

The password is not an actual field in the database. When the user types in their password, the first name and last name are parsed out on the "*" using VBScript. The database is searched for the company associated with that combination of first and last name. The danger here is if two contacts from separate companies have identical first and last names. We checked the database and made sure that there were no doubles. There were several sets of duplicated first name/last name combinations. On closer inspection, however, each of these was found to be a data-entry error or an outdated

contact record. We culled out the erroneous entries so the passwords would work correctly.

If, in the future, more than one contact has the same first and last name, it will be necessary to come up with an alternative solution for the password. One possibility would be to use the contact person's first and last name and add the invoice number. The invoice number, as mentioned earlier is a number assigned by the UCS office that is used for billing purposes.

Once the password question was answered, the forms were created using ASP with server-side VBScript to connect to and query the database. Some of the client-side functionality was created using JavaScript since client-side VBScript is not compatible with Netscape Navigator. In order to maintain the data throughout the pages, I stored the information from each form in Session variables with server-side VBScript.

Walkthrough of Employer Interface

All Screenshots mentioned in this section can be found in appendix G. As mentioned earlier in the "Users of the Proposed System" section, the employer will be sent an invitation in the mail which lists the url for the online registration form and the employer's password for accessing the form. When the employer types the url for the registration form into their web browser, they are presented with the login screen. After the employer types in their password and hits the submit button, the password is parsed on the asterisk (*) by a block of VBScript code as mentioned earlier. The code is listed below:

```
<%
'save password from form into Session variable
```

```
Session("password") = Request.Form("password")

Dim strPassword, intPostion, intSize, strLname, strFname
strPassword = Session("password")
'Find the asterisk in the password
intPosition = InStr(strPassword,"*")

'The purpose of this code is to transfer the firstname and last name
'  portion of the user-supplied information to a variable for later
'  use in the query titled query1.
if intPosition = 0 then
   Response.Write "The password information is not in the
   expected format."
else
  'Determine how long the last name portion is based on where the
  '  asterisk is and the overall length of the string
    intSize = len(strPassword) - intPosition

  'Assign the last name string to a variable
    strLname = mid(strPassword, intPosition+1, intSize)
    strFname = mid(strPassword, 1, intPosition-1)

end if

Session("strFname") = strFname
Session("strLname") = strLname
%>
```

The code above pulls the first and last name of the contact out of the password string entered by the user and stores the resulting strings in variables that are used to query the database. The SQL query that is sent to the database is as follows:

```
<%
query1 = "SELECT org_id, org_name, website, nonprofit, "
query1 = query1 & "fname, lname, address, "
query1 = query1 & "city, state, zip, "
query1 = query1 & "phone, fax, email, event_code,
invoice_num, datestamp "
query1 = query1 & "FROM Organizations WHERE "
query1 = query1 & "fname="
query1 = query1 & "'" & Session("strFname") & "'"
query1 = query1 & " AND lname="
query1 = query1 & "'" & Session("strLname") & "'"
```

```
'following code for testing query
'Response.Write("query1 is: " & query1)

'Open the recordset
  objRS.Open query1,"DSN=FairDB.dsn",adOpenStatic
'Set event code and invoivce number as Session variables for use on confirmation
page
Session("event_code") =  objRS("event_code")
Session("invoice_num") =  objRS("invoice_num")
%>
```

The resulting page is the page titled "Contact Information". The employer is presented with the name of their organization, which cannot be edited. They are also presented with input fields for contact first name, contact last name, address, city, state, zip, phone, fax, email, website address, and a yes/no checkbox to determine if their company is non-profit. These fields are populated with current data from the database, and employers are given the opportunity to edit these fields. Required fields are marked with an asterisk. If the user fails to enter a value for any of these required fields before they hit the submit button, a Javascript block of code causes an alert box to pop up asking them to enter a value for whichever field is empty.

Once the user hits the "Continue" button on the "Contact Information" page, they are presented with the "Registration Fees" page. The non-profit field from the previous page determines the layout and content of this page. If the value of the non-profit field is yes, the user is presented with the reduced fees for non-profit organizations. If the non-profit field is no, the user is presented with the regular fees along with a choice to take advantage of the all-fair discount .

On this page, the user enters the number of representatives who will be attending each fair. Once they have entered their information, they can click the "Calculate Fees" button to view the subtotals for individual fairs, and the total for all the fairs combined. When

they click the "Calculate Fees" button, a Javascript block of code is run to validate the data. If more than one representative is entered for the 1st representative field for any of the fairs, or if more than one representative is entered for the 2nd representative field for the minority night, then an alert box is displayed which asks the user to correct the erroneous data.

If the user does not click the "Calculate Fees" button before clicking the "Continue" button, the fees will still be calculated so that the values passed to the subsequent page will be correct. The validation script that is run when the "Calculate Fees" button is clicked also is run when the "Continue" button is clicked.

The next page the user is presented with is titled "Meals & Parking". On this page, the user enters the number of lunches they would like for the Technology and Carolina Career Fairs, whether or not they need parking, and any accommodations they require to participate in the fairs. This information is not pulled from the database since it will be blank before the registration form is filled out. Once the user has entered their information on this page, they hit the "Continue" button and are presented with the "Student Lists" page.

On the "Student Lists" page, the user can check off which lists they would like to receive. Each list costs six dollars. Once the user checks the lists they would like to receive, they can click on the "Calculate Total for Lists" button to view the amount owed for lists, or they can simply click the "Continue" button to view the next screen. If they do not click the "Calculate Totals" button before clicking the "Continue" Button, the totals will still be calculated so that the correct values will be passed to the next page.

The next page the user is presented with after the "Student Lists" page is the "Publicity Information" page. This page allows the user to enter a brief description of their organization and other biographical information that will appear on handouts for students and be displayed on the online student search page. The "Organization Description" and "Industry" fields are required. Users are also required to check at least one box in the category "Majors Considered". The fields are initially populated with information from the database. This information will be the information that was entered the previous year. If the information has not changed, the user can simply click "Continue" and move to the next screen. Otherwise, they can edit the information and then proceed to the next screen.

The "Confirmation Page" page of the input form displays all the data that the user entered on the previous screens. The user is given the choice to go back and edit certain sections of the form if they need to make changes. Once they are satisfied with the data they have entered, they are instructed to print 2 copies of the final page, one copy to keep and one to mail in with their registration money. Then they can hit the "Submit Registration" button to submit their registration. At this point the update query is sent to the database to update their information. After they have submitted their registration, they are presented with a "Thank You" page. If they try to fill out the registration form a second time, they are presented with a message asking them to contact the Career Services Office to make any changes to their registration information.

We considered online payment, but university policy does not allow it. The alternative we came up with, as described above, is to allow users to enter their data online then print out the form and send it in with their payment. The Secretary in charge

of the fairs uses the Career Services administrative interface to the database (described in the next section) to enter the date payment was received and the amount received. There is also a yes/no field for registration confirmation on the administrative side. Once the total payment has been received and the registration form approved, the confirmation box is checked and the company is officially registered.

Career Services Administrative Interface

The administrative interface for Career Services was designed in Access. An input form was designed for data entry (appendix H). The people who need to access the database in Career Services must first map a network drive to the server housing the database so that they will be working with the current copy of the database. Once the forms are returned in the mail with payment, the Secretary in charge of the fairs checks the information in the database to make sure it matches the information on the form. She then enters the date the form was received, the amount received for the fairs, and the amount received for the lists. If the total amount owed has been paid in full and there are spaces available for the fairs the company has registered for, she marks the company as confirmed. She then runs a report that creates a confirmation letter that she mails to the company.

Student Web Interface Design

The next step was to create the web pages that students could use to search the database. The purpose of the student interface was to give students the ability to view a list of companies coming to each fair and a list of companies by Industry. There were

concerns about how well Access would perform if many students tried to search the database simultaneously.

A decision was made to create the pages as dynamic web pages that would pull from the database each time a student viewed the page. The advantage of this solution is that it is up to date with the information in the database at the time that the student is viewing it. The disadvantage of this solution is the fact that it can cause unacceptable performance degradation if the traffic passes a certain point. The student interface was modeled after the current page, an example of which can be found at http://www.unc.edu/depts/career/FallCareerFairs/consult1.html (appendix I).

Implementation

Criteria for Success

The measure of success for the Access database is whether or not it accurately models the data that Career Services needs to gather and whether it can support the necessary queries and supply the necessary reports. The employer interface will be considered a success if it is easy to use and understand by the representatives who will use it to enter their registration information. It must also provide data-validation functionality in order to maintain the integrity of the database. A crucial measure of success, of course, is if employers can and do use the web interface. The measure of success for the student interface is how easy it is to use to search the database for which companies are coming to which fair and to search for companies by industry. Another

important consideration for the student interface is that the pages not cause an unacceptable degradation of performance.

Evaluation

In order to evaluate the Access database, I met with the Secretary in charge of the three career events. She is the person who will do the majority of the data entry. Since migrating to any new information system can be stressful, it was very important that she feel comfortable with the main data-entry form and that she was confident she could find the information she needed when she needed it.

In order to test the input form, the Secretary and I sat down together and I watched as she used the form to enter data from some of the first registration forms that had been returned for the coming Career Fairs. The Career Services Office had divided their mailing into three groups. The first mailing was sent to seven companies on June 2, the second to one hundred companies on June 16, and the third to approximately six hundred companies on June 30.

The companies in the first two mailing groups were sent hard copies of the registration form to be filled out and returned with their payment. The third mailing group was given a password so they can access the online database and enter their information online. The reason for splitting the mailings is to give priority to certain companies based on criteria decided upon by the Career Services Office. We decided to use the registration forms that were returned from the first two mailings to test the Access data-entry form as well as the online form. This would allow us to work out any problems that may occur before the bulk of the employers access the online database.

I talked with the Secretary as she used the data-entry form in Access to find out how comfortable she felt with it and to determine if any final modifications needed to be made. She had no problems entering the data and said she was satisfied with the design.

The next step was to evaluate the employer's online interface. In order for the employers at Career Services to evaluate the online forms, I created temporary entries for them in the database. They were then able to go online and enter information in the database. I collected their feedback and made modifications accordingly.

The criteria for success for the student interface was whether the interface was easy to use and whether performance was satisfactory once students began using it. At the writing of this paper, it was too soon to find out how the system would withstand the actual traffic the database will get when students begin accessing the page in August. Because of this limitation, an alternative solution was devised to deal with the possibility of performance degradation (see section titled "Load Concerns").

Lessons Learned

Looking back on the design and development process, one of the biggest lessons I learned is the importance of creating a detailed requirements document before any design work is begun. Having the design decisions in writing and having the client sign off on a requirements document would help to avoid misunderstandings during development. An example of where a requirements document would have helped in the current project is in regards to the password problem. Instead of spelling out my plans for creating a random seven-digit password for employers, I assumed that this was a detail that would not be

important to discuss with the client. I spent a good deal of time devising a Perl script that would create the password and migrating the created passwords into the Access database.

Only after I had created the passwords and populated the database with them, did I find out that the client did not want random passwords that would be difficult for the employers to remember. If I had proposed the solution in an initial requirements document, I would have saved myself the time spent writing the Perl script. The lesson I gained from this is that a very detailed requirements document should be written and signed off on by the client before any labor goes into the actual design and development work.

Another lesson that was reinforced with this project was that it is vital to test all pages in both Microsoft's Internet Explorer and Netscape's Navigator on several different machines. If possible, it is a good idea to test on a Macintosh and an IBM on several different screen sizes and with several different versions of the browsers mentioned above. This project reminded me of this golden rule after I had coded several pages with client-side VBScript only to find out that client-side VBScript is not supported by Netscape Navigator. The book I had been using for reference focused on Microsoft solutions and failed to mention the fact that many of the examples in the book were not supported in Netscape. So for server-side scripting I used VBScript, but for client-side I switched to JavaScript which is supported by both browsers.

Deployment Plans

The plans for deployment are to send out invitations with access instructions to employers. The employers will then be able to access the online database and enter their

information. For the first several weeks after the mailing is sent, employees at Career Services will call a few of the employers who register online and ask for their feedback on the online form. In addition to this, someone in Career Services will run a query that will list the employers who have registered for that day. Those employers will be sent an email asking for their feedback on the online registration form. This feedback will be used to make modifications to the form.

The Student Interface will be made available in early August. Students will be provided with an email link to send any feedback they have about the search interface. Server load and performance will also be monitored. If the performance falls below an acceptable level, the alternative solution to dynamic pages discussed earlier in this paper will be carried out.

Expected Modifications

During the original information gathering session, the Career Services employees in charge of the career fairs said that there was not a need to have more than one contact associated with an organization. On further consideration, they came to the conclusion that it would be a useful feature to be able to associate more than one contact with an organization. This will require a modification of the underlying structure of the database.

In order to provide this functionality, a new contact table will need to be created. It will have a one-to-many relationship with the organization table. The organization table will be on the "many" side of the relationship with org_id as a foreign key in the contact table, tying it to the organization table.

Another modification that will be necessary is to research and decide on the best location for the event code and invoice_num. These fields were added after the design of the database and very near the date of implementation. In the final Excel spreadsheet I was given to migrate to the database, the invoice number and event code were tied to the contact name. It will take further investigation to decide where the best location is for the invoice code once the database is altered to accommodate more than one contact per organization.

Another modification that would enhance the functionality of the online database is allowing students to search the database by major or job title. With the current solution, they are able to search by event and industry. These search fields were chosen because they were the ones that were currently offered and the students were accustomed to searching on them. The next step would be to add these new fields once the students accept and become comfortable using the new version of the database.

In addition to enhancing the student interface, it is expected that new career fairs will be added to the database if the system is a success this year. The goal is to migrate the registration information for all the career events sponsored by the University Career Services office to this online database.

The final modification that is possible is migrating the backend database to SQL Server. As mentioned earlier, this is seen as a last resort due to the expense entailed in training Career Service personnel to use and maintain the database.

Conclusion

The purpose of this project was to streamline the data management process associated with the three career fairs sponsored by the UNC Career Services Office each fall. In order to achieve this, an internet-accessible relational database was created. The Career Services Office is a small office with very specific needs. The relational database facilitates internal use of the data by Career Services employees. Adding web functionality to the database allows students to view information and employers to enter their own information online. This has the potential to ease the burden on Career Services staff by shifting the responsibility for data-entry to employers. The time required to manually create static search pages for the students is also freed up by the dynamic web pages that pull their information from the database.

The online database implemented in this project is an example of leveraging database and web technologies to solve a problem that is very common, especially in small, under-staffed offices. This problem arises when the amount of data collected reaches a certain size and level of complexity. To deal with the growing amount of data, office staff often spend more time entering and tediously retrieving this data than they spend actually carrying out the tasks they need the data to accomplish. By providing a more efficient method of data input and retrieval, a web database allows office staff to concentrate on the tasks at hand.

Selected Bibliography

Brophy, Keith and Koets, Timoty. *Teach Yourself VBScript in 21 Days.* Indianapolis: Sams.net Publishing, 1996.

Elmasri, Ramez and Navanthe, Shamkant. *Fundamentals of Database Systems 2nd Edition.* New York: Addison-Wesley Publishing Company, 1994.

Homer, Alex, Sussman, Dave, and Francis, Brian. *Professional Active Server Pages 3.0 .* Birgmingham, UK: Wrox Press, 1999.

Kauffman, John. *Beginning ASP Databases.* Birgmingham, UK: Wrox Press, 1999.

MacFarlane, Nigel et al. *Professional Javascript.* Birmingham, UK: Wrox Press, 1999.

Prague, Cary N. and Irwin, Michael R. *Microsoft Access 97 Bible.*